MULTIMEDIA UNIVERSITY®

STUDENT ID NO

MULTIMEDIA UNIVERSITY

FINAL EXAMINATION

TRIMESTER 2, 2017/2018

ECP1026 – ALGORITHM AND DATA STRUCTURE

(TE/RE)

8 MARCH 2018
2.30 p.m. – 4.30 p.m.
(2 Hours)

INSTRUCTION TO STUDENTS

1. This Question paper consists of 7 pages including cover page with 4 Questions only.

2. Attempt **ALL** questions. All questions carry equal marks and the distribution of marks for each question is given.

3. Please write all your answers in the Answer Booklet provided.

Question 1

(a) Consider the Finite State Automata (FSA) in Figure Q1.1.

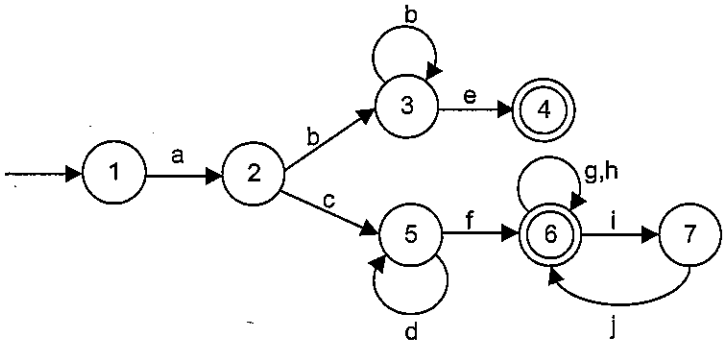

Figure Q1.1

(i) Identify the set of states Q, set of input symbols Σ, initial state q_0 and set of final states F of the above FSA. [4 marks]

(ii) Express the above FSA with a regular expression (RE). [6 marks]

(b) Describe the components of a Turing Machine, with the aid of diagrams. [6 marks]

(c) Consider the Turing Machine (TM) given in the Figure Q1.2.

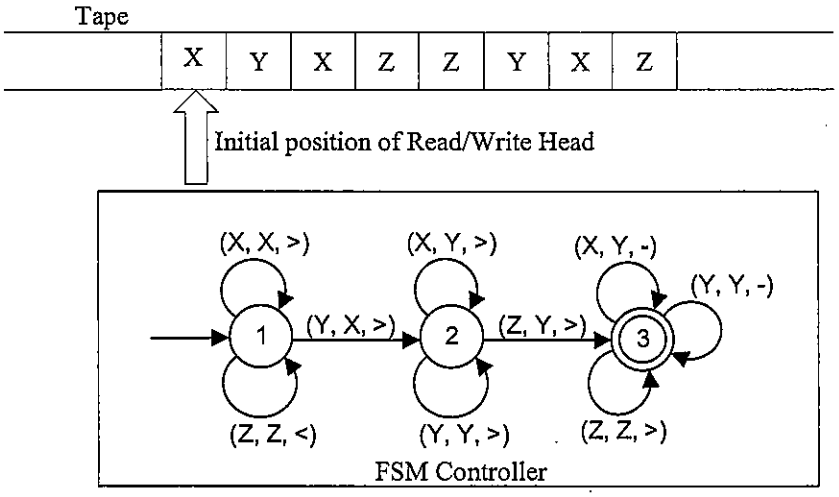

Figure Q1.2

(i) Run the TM on the given input tape. Determine the output on the tape at the end of the execution. [8 marks]

(ii) Indicate position of the read/write head at the end of the execution. [1 mark]

Continued...

Question 2

(a) A simple car insurance premium can be calculated with the sum insured (based on the car's market value) and no-claim discount (NCD). It is calculated by the formula,

$$\textit{Car insurance premium} = (\textit{comprehensive rate for first RM 1,000 sum insured} + \textit{RM 26 for each RM 1,000 or part thereof on value exceeding the first RM1,000}) \times (100 - \textit{NCD})\%$$

where *comprehensive rate* varies based on car's cubic capacity (*cc*), as in Table Q2.1; *NCD* refer to discount that car owner can enjoy over the coverage duration that he/she does not claim any damages from the insurance company, as in Table Q2.2.

For example, for a 1,500 *cc* car with sum insured RM 50,000 and 1 year of NCD coverage duration, its *comprehensive rate* is RM 305.50, and its car insurance premium is calculated as:

$$(\text{RM } 305.50 + \text{RM } 26 \times 49) \times (100 - 25)\% = \text{RM } 1{,}184.63$$

Cubic Capacity (*cc*) Not Exceeding	*Comprehensive Rate* (RM)
1,400	273.80
1,650	305.50
2,200	339.10
3,050	372.60

Table Q2.1

NCD Coverage Duration (year)	*NCD* (%)
1	25
2	30
3	38.33
4	45
5 and above	55

Table Q2.2

Write a complete C program consisting of a user-defined function called `simpleCarPremium`. The program should perform each of the following tasks:

- prompt user to input for sum insured, car's cubic capacity and NCD coverage duration in year,
- function `simpleCarPremium` to calculate and return the car insurance premium,
- print the car insurance premium on the monitor screen.

A sample program output is given below (texts in *italic* are user inputs):

```
Enter sum insured, car's cubic capacity and NCD coverage
duration in year: 50000 1500 1
Car insurance premium is RM 1184.63
```

[12 marks]

Continued...

(b) For a certain binary tree,
- the preorder traversal is PQSUVTR,
- the postorder traversal is UVSTQRP,
- the inorder traversal is USVQTPR.

Draw the binary tree that fulfils these conditions. [7 marks]

(c) For the graph given in the Figure Q2.1, represent it as data structures in the following ways:
(i) Adjacency matrix [4 marks]
(ii) Set of pairs [2 marks]

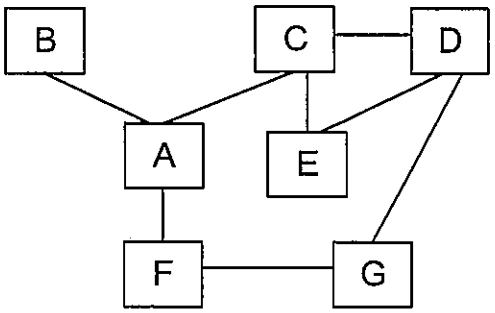

Figure Q2.1

Continued...

Question 3

(a) Addition on two integers x and y (where $x \geq 0$) can be performed by repetitively decrementing x by one and at the same time incrementing y by one, until x reaches 0, and the final value of y will be the result of the addition.

For example, if $x = 2$ and $y = 5$, by substitution,

$$\begin{aligned} f(2,5) &= f(1,6) \\ &= f(0,7) \\ &= 7 \end{aligned}$$

(i) Express the above-mentioned addition method with mathematical formulae in terms of inputs x and y. The mathematical formulae must consist of both the formulae for the base case and the recursive case. [4 marks]

(ii) Implement the mathematical formulae that you have constructed in part (a)(i) in ANSI-C using recursion. The function prototype is given as:

```
int f(int x, int y);
```

[6 marks]

(b) Study the recursive function given in Figure Q3.1.

```
void fill(int row, int col) {
  if(map[row][col] != ' ')
    return;

  map[row][col] = c++;
  fill(row,col+1);
  fill(row,col-1);
  fill(row-1,col);
  fill(row+1,col);
}
```

Figure Q3.1

The global character variable `c` has been initialized to `'a'`. Illustrate the final contents of the character array `map` given in Figure Q3.2 following the execution of a function call `fill(1,2)`.

col / row	0	1	2	3	4	5
0	X	X	X	X	X	X
1	X				X	X
2	X	X	X			X
3	X		X		X	X
4	X	X	X	X	X	X

Figure Q3.2

[5 marks]

Continued...

(c) Consider the C code given in Figure Q3.3. Indicate clearly the time complexity of each statement, if any, and determine the worst-case time complexity and Big-O of the program. [7+2+1 marks]

```
1   #include <stdio.h>
2   int main(){
3     int n, k1, k2, x;
4
5     printf("Enter number of data: ");
6     scanf("%d", &n);
7
8     for(k1=1; k1<=n; k1++){
9       if(k1%2){
10        x=0;
11        for(k2=1; k2<=n; k2++)
12          x = k1*x + k2;
13        printf("%d: %d\n", k1, x);
14      }
15      else {
16        printf("%d: %d\n", k1, x=0);
17      }
18    }
19    return 0;
20  }
```

Figure Q3.3

Continued...

Question 4

(a) Write the pseudocode of the iterative implementation of binary search algorithm applied on an array. [6 marks]

(b) State what collision is in the context of search, and briefly explain three collision resolving methods. [7 marks]

(c) Figure Q4.1 shows an incomplete insertion sort function that arranges a list of integer values in ascending order.

```
void insertionSort(int a[], int n){
  int k1, k2, k3, temp;

  for(k1=0; k1<n; k1++){
    temp = a[k1];

    //Set k2 to k1
    //While k2 is greater than 0 AND array element k2-1 is ...
    //    ... greater than temp
    //  Shift the array element k2-1 to position k2
    //  Decrement k2 by one
    //Place temp at position k2

    printf("\n%2d: ", k1);
    for(k3=0; k3<n; k3++)
      printf("%3d", a[k3]);
  }
}
```

Figure Q4.1

i) Based on the comments given in Figure Q4.1, complete the insertion sort function. [6 marks]

ii) Write the formatted program output when the insertion sort function is executed on an array containing elements: 20, 11, 33, 42, 57, 32, 62 and 68. [4 marks]

iii) Explain briefly in what situation insertion sort exhibits the best-case behavior. [2 marks]

End of Paper